Understanding CSS Essentials: Content Flow, Positioning, and Styling

3.1. Understand the core CSS concepts.

Agenda

1 Cascading Style Sheets

2 Selectors and Declarations

3 Fonts and Font Families

4 Content Flow

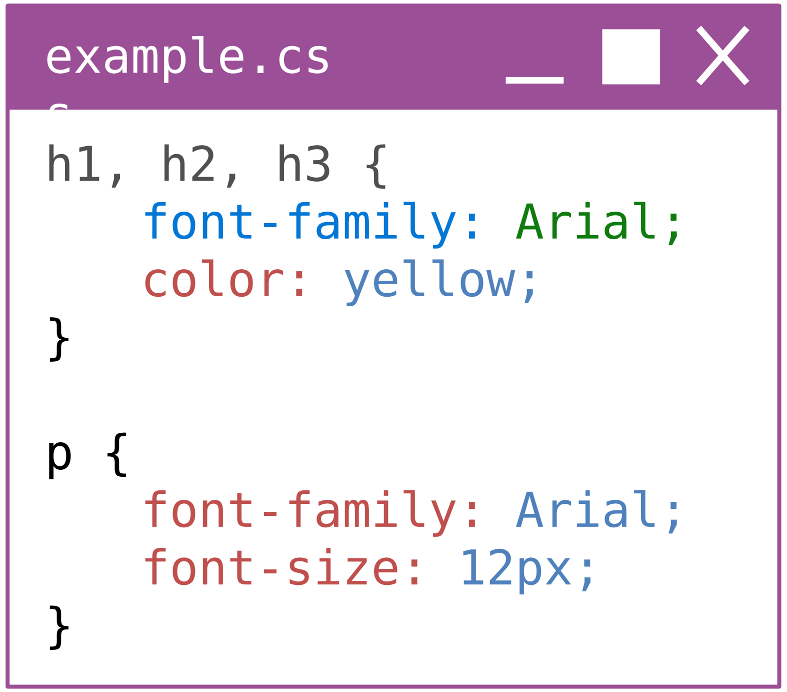
5 Positioning

6 Content Overflow

Cascading Style Sheets

Cascading Style Sheets (CSS)

- **Cascading Style Sheets (CSS)** is a language that defines how HTML elements are styled
  - In other words, HTML structures a document, while CSS formats it
- CSS can be applied to single HTML elements, or it can be kept in a separate file and applied to elements throughout a document
  - Files that hold CSS are called “style sheets” and use the .css exension
- Unlike HTML, CSS uses **rules** instead of tags



```
h1, h2, h3 {
    font-family: Arial;
    color: yellow;
}

p {
    font-family: Arial;
    font-size: 12px;
}
```

CSS3

- CSS3 is the version of CSS that was released in conjunction with HTML5
- CSS3 is backward compatible, which means it can be used with older versions of HTML and CSS
- There are a number of new effects in CSS3, including 2D and 3D transformations, as well as animations

Linking CSS with HTML

There are several ways to incorporate CSS into HTML:

1. inline styling
2. use of the <style> element nested in the <head> element
3. the <link> element and a separate CSS file

Placing CSS in a separate file has a lot of advantages:

- A style change can apply to an entire document
- Teams working on Web projects can separate responsibilities

1
```
<p style="color: red">
```

2
```
 <style>
   h1 {
     font-family: 'Segoe
UI';
     color: #808080;
   }
 </style>
```

3
```
 <link href="StyleSheet.css”
         rel="stylesheet"
     type="text/css">
```

How do you link HTML and CSS files?

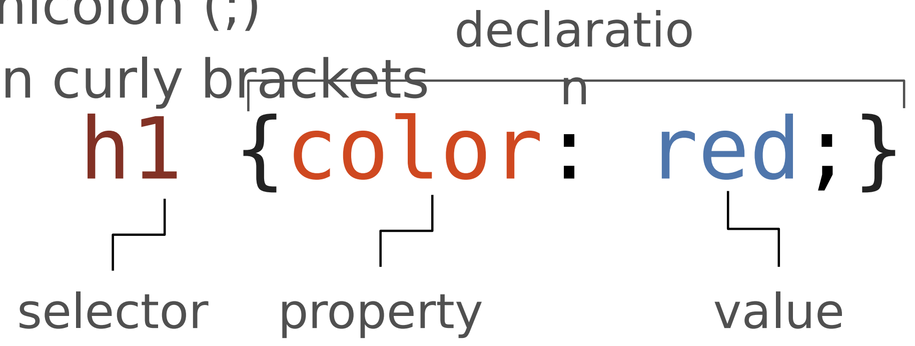
- CSS files are linked to HTML files using the `<link>` element
- The `href` attribute points to the location of the CSS file
  - It's <u>extremely</u> important that the file name and location are correct or none of the style will be applied
- The `rel` attribute should be set as "stylesheet", while the `type` attribute should be set as "text/css"

```
<link href="StyleSheet.css" rel="stylesheet" type="text/css">
```

Selectors and Declarations

Selectors and Declarations

- There are two parts to a CSS rule:
  - selectors
  - declarations
- A **selector** references the HTML element that you want to style
- A **declaration** is the style that you want to apply to that element
  - declarations have two parts: a property followed by a colon (:) and a space, and a value followed by a semicolon (;)
  - declarations sit between curly brackets



```
h1 {color: red;}
```

IDs and Classes

- HTML elements can be referenced by selectors in a number of ways, including:
  - the tag name, such as p, h1, table, etc.
  - ID selectors, such as #navbar, which include a hashtag symbol (#) as a prefix
  - Class selectors, like .happy, which include a decimal (.) as a prefix
- id and class are both universal attributes
  - id is used to identify unique elements
  - class should be used to categorize elements into groups that will be styled similarly

tag name

```
h1 {
    color: red;
    font-family: sans-
serif;
    text-align: left;
}
```

id selector

```
#navbar {
    background-color:
green;
}
```

class selector

```
.happy {
    font-size: 14px;
}
```

Comments

- Add comments to your CSS file to explain what it does
  - It will also make your code easier for others to read and understand
- Add a multi-line comment by starting with the /* characters and ending with the */ characters
  - Anything between these characters won't be interpreted by a browser

```
/*These comments are typically
reserved for making multi-line
comments*/

p {
   color: red;
   /* This is a single-line comment */
   text-align: center;
}
```

Fonts and Font Families

Fonts and Font Families

- A font is a set of characters of a particular size and style
- The primary way to specify fonts in CSS is by using the `font-family` property
- Three common types of fonts are serif, sans serif, and `monospace`

Font Type	Example	Description
Serif	Times New Roman	feature decorations at the ends of certain characters
Sans Serif	Arial	features no decoration at the ends of characters
Monospace	`Courier`	each letter occupies the same amount of screen space

The `@font-face` rule

Prior to CSS3, developers had to use fonts that were installed on a user’s device

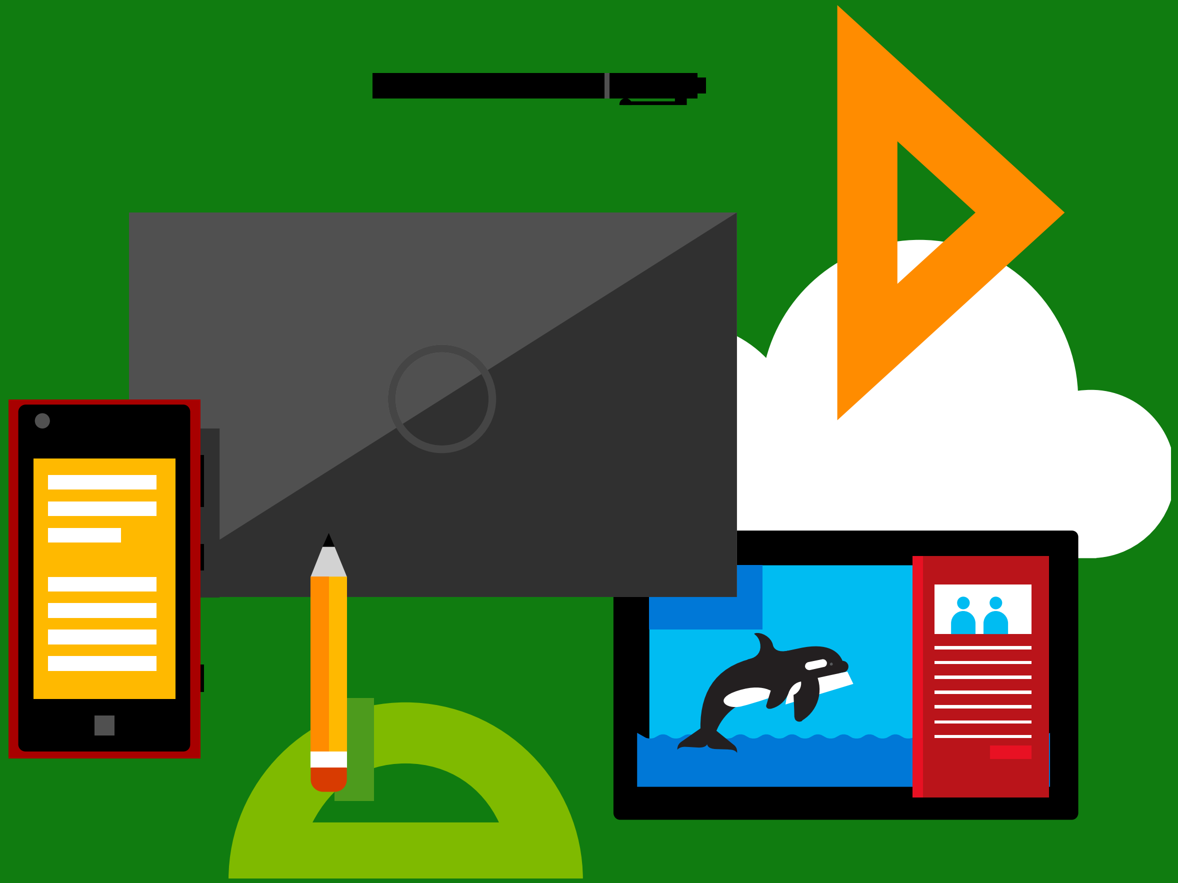
- these fonts are known as web-safe

This greatly limited the number of font choices available to developers

The `@font-face` rule in CSS3 allows developers to use any font as long as it is located on their Web server

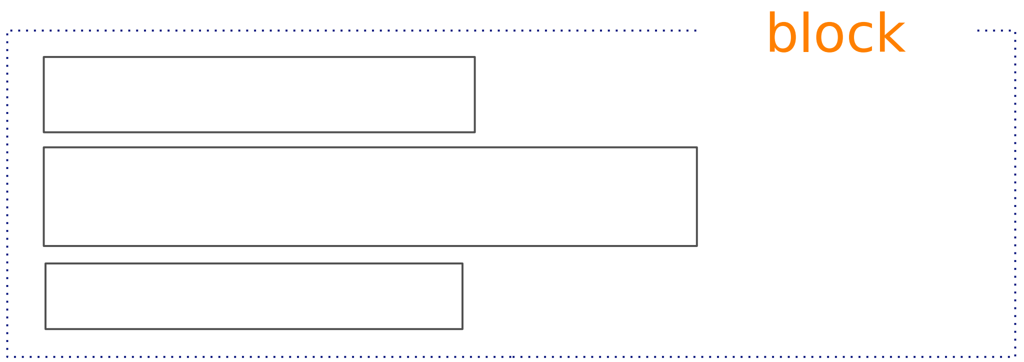
```
@font-face {
    font-family: "font-family-name";
    src:
url("http://website/fonts/fontfile")
}
```

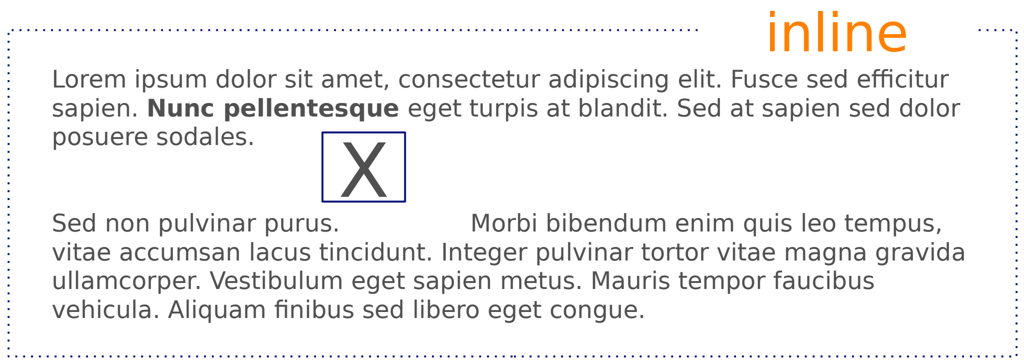
Content Flow



Content Flow

- **Flow** is a display style in HTML that focuses on filling content on a page horizontally in rows from top to bottom
- There are two different types of flow:
  - Block
  - Inline
- With **block** flow, content is placed on its own line above or below other content
  - It fills the line from left to right
- With **inline** flow, content fits on the same line with other content before or after it

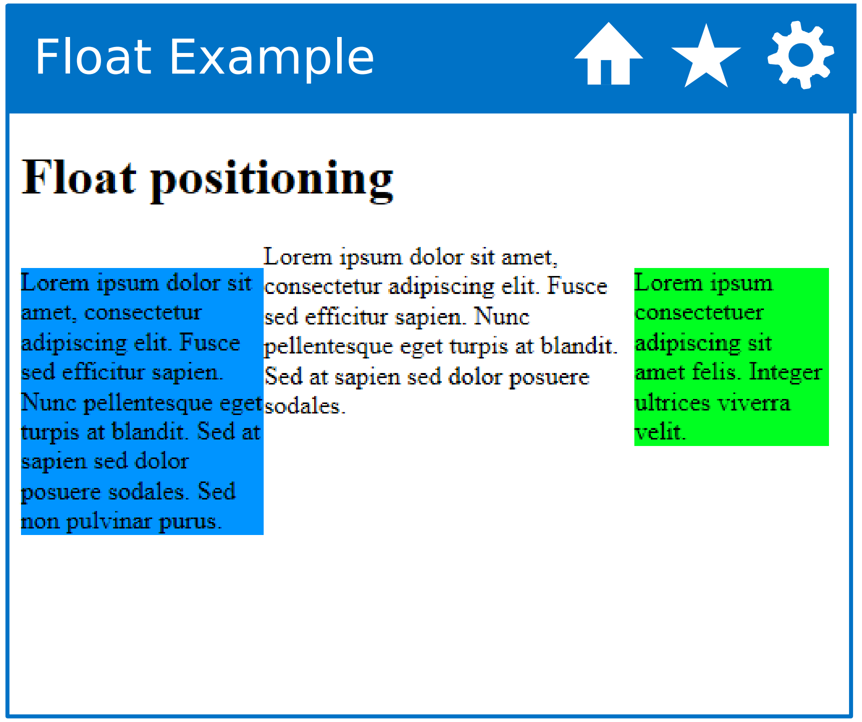




Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce sed efficitur sapien. **Nunc pellentesque** eget turpis at blandit. Sed at sapien sed dolor posuere sodales.

Sed non pulvinar purus. Morbi bibendum enim quis leo tempus, vitae accumsan lacus tincidunt. Integer pulvinar tortor vitae magna gravida ullamcorper. Vestibulum eget sapien metus. Mauris tempor faucibus vehicula. Aliquam finibus sed libero eget congue.

Positioning


Float Example

Float positioning

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce sed efficitur sapien. Nunc pellentesque eget turpis at blandit. Sed at sapien sed dolor posuere sodales. Sed non pulvinar purus.

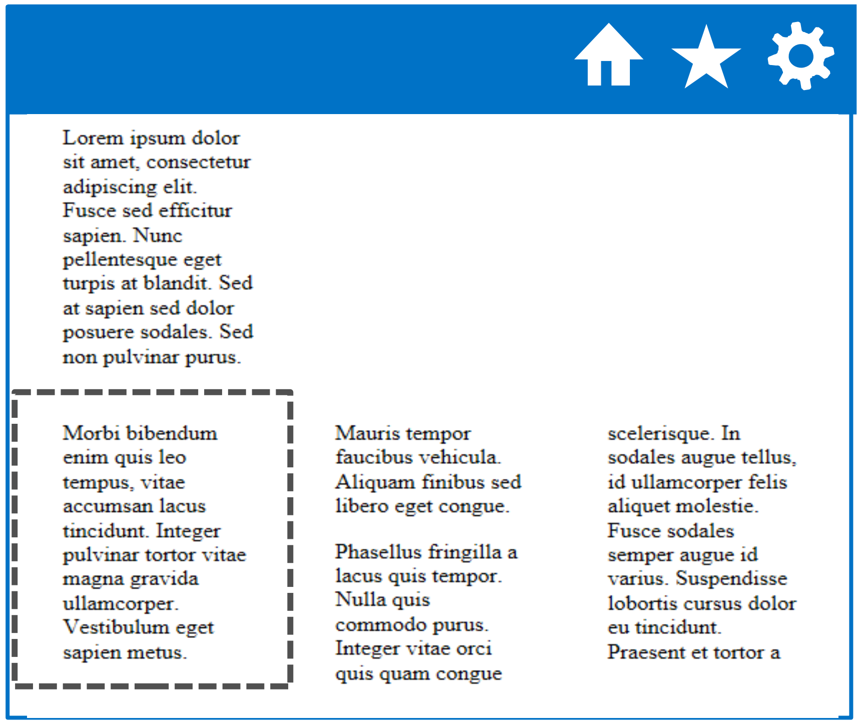
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce sed efficitur sapien. Nunc pellentesque eget turpis at blandit. Sed at sapien sed dolor posuere sodales.

Lorem ipsum consectetuer adipiscing sit amet felis. Integer ultrices viverra velit.

Floating Elements

- In addition to modifying the flow of a page, HTML and CSS allow developers to position individual HTML elements
- The **`float` property** allows you to move an element all the way to the left or right of a page
  - other content will wrap around the floated content
- The value of `float` is commonly set to `left` or `right`

Clearing a Float


Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce sed efficitur sapien. Nunc pellentesque eget turpis at blandit. Sed at sapien sed dolor posuere sodales. Sed non pulvinar purus.

Morbi bibendum enim quis leo tempus, vitae accumsan lacus tincidunt. Integer pulvinar tortor vitae magna gravida ullamcorper. Vestibulum eget sapien metus.

Mauris tempor faucibus vehicula. Aliquam finibus sed libero eget congue.

Phasellus fringilla a lacus quis tempor. Nulla quis commodo purus. Integer vitae orci quis quam congue

scelerisque. In sodales augue tellus, id ullamcorper felis aliquet molestie. Fusce sodales semper augue id varius. Suspendisse lobortis cursus dolor eu tincidunt. Praesent et tortor a

- Use the **`clear` property** to prevent other floating elements from touching the left or right hand sides of an element
- The value of the `clear` property can be `left`, `right`, `both`, or `none`.
  - `left` clears the left side
  - `right` clears the right side
  - `both` clears both sides
  - `none` allows other elements to touch

Float Demo

```
<body>
    <h1>Positioning Demo</h1>
    <div id="block1">
       <p>Hi! I'm a blue block.</p>
    </div>
    <div id="block2">
       <p>Hi! I'm a green
block.</p>
    </div>
</body>
```

```
<style>
         #block1 {
             width: 200px;
             background-color:
   blue;
             padding: 10px;
         }

         #block2 {
             width: 200px;
             background-color:
   blue;
             padding: 10px;
         }
     </style>
```

Positioning Elements

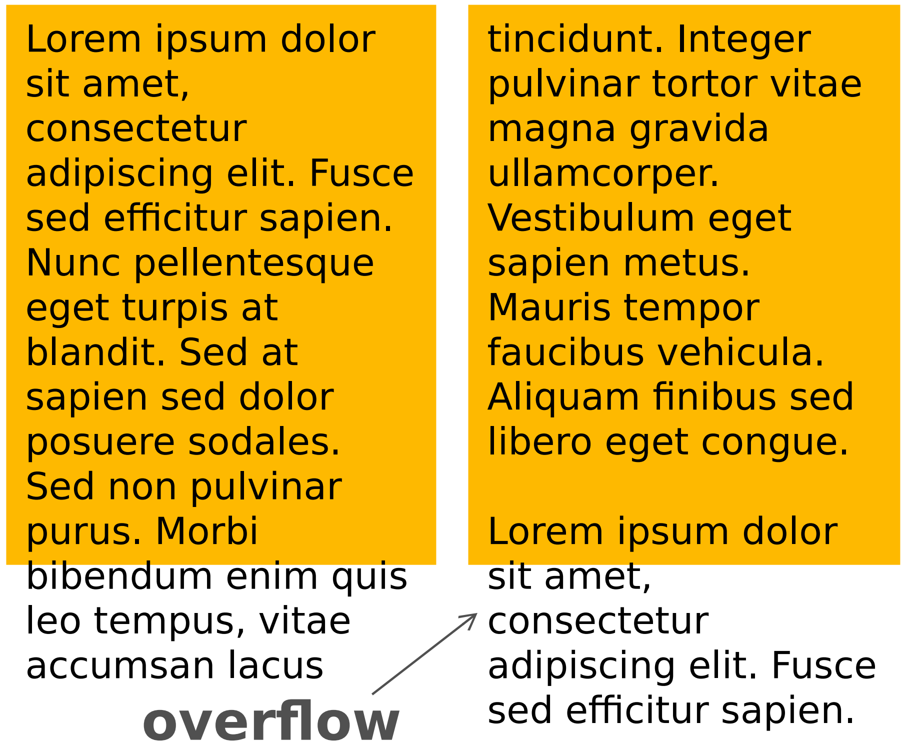
- The **`position` property** allows you to place an element at specific locations within a document
- The values commonly used with the `position` property are

VALUE	DESCRIPTION
`static`	positions an element within its normal flow
`relative`	positions an element in relation to where it would normally be positioned in the flow of content
`absolute`	ensures that the positioning of an element doesn’t impact other content
`fixed`	positions an element in relation to the browser window and stays in the same place

Content Overflow

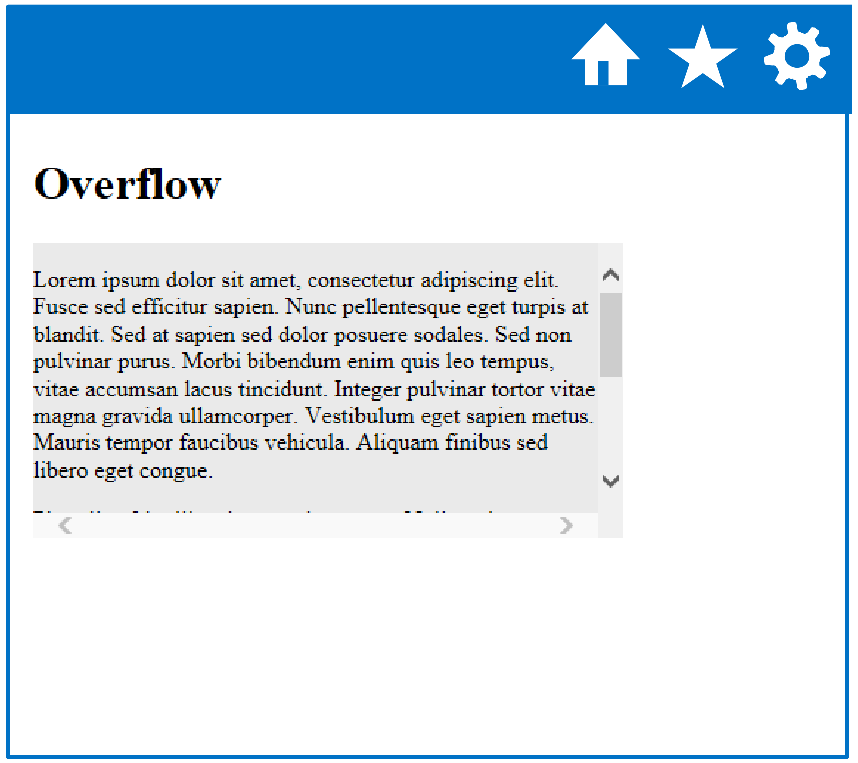
The Bounding Box

- Every HTML element on a page occupies a rectangular space called a **bounding box**
  - Think of text with a highlighted background on a Word document or web page
- With CSS, you can modify the height or width of the bounding box
- If an element doesn't fit inside of its bounding box, then that content is called **overflow**
- You can modify how the overflow content is styled using the `overflow` property in CSS


Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce sed efficitur sapien. Nunc pellentesque eget turpis at blandit. Sed at sapien sed dolor posuere sodales. Sed non pulvinar purus. Morbi bibendum enim quis leo tempus, vitae accumsan lacus
overflow
tincidunt. Integer pulvinar tortor vitae magna gravida ullamcorper. Vestibulum eget sapien metus. Mauris tempor faucibus vehicula. Aliquam finibus sed libero eget congue.
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce sed efficitur sapien.

Understanding Scrolling Overflow

- If you set the `overflow` property's value to `scroll`, then all of the content in an element will stay within the bounding box
  - none of the overflow will appear outside of the box
- This will allow users to use scroll bars within the box to view all of the content
  - this is the same type of scrolling you experience on a Web page or in a Web app

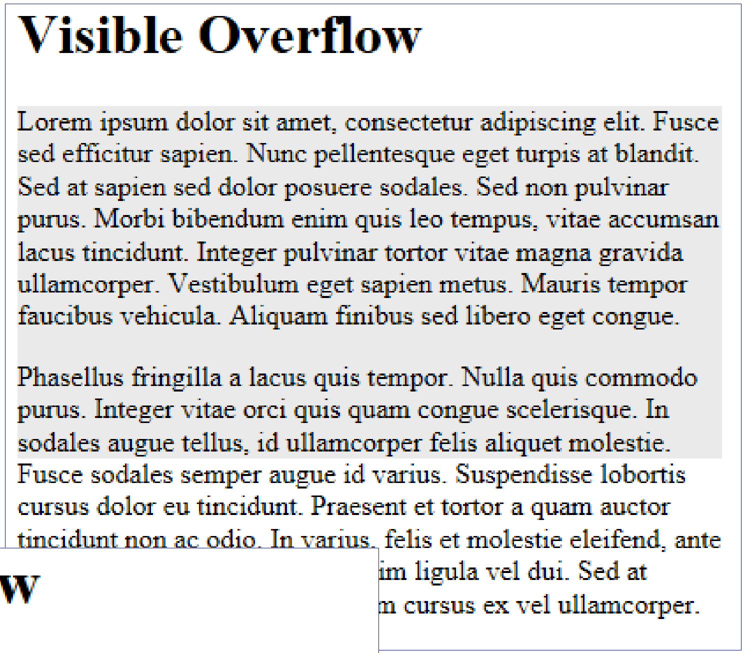


Overflow

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce sed efficitur sapien. Nunc pellentesque eget turpis at blandit. Sed at sapien sed dolor posuere sodales. Sed non pulvinar purus. Morbi bibendum enim quis leo tempus, vitae accumsan lacus tincidunt. Integer pulvinar tortor vitae magna gravida ullamcorper. Vestibulum eget sapien metus. Mauris tempor faucibus vehicula. Aliquam finibus sed libero eget congue.

Understanding Visible Overflow and Hidden Overflow

- With visible overflow, overflow content will spill outside of the bounding box and potentially overlap with other elements
  - implement visible overflow by using the `overflow` property's visible value
- In contrast, hidden overflow will make any content outside of the bounding box invisible
  - implement `hidden` overflow by using the overflow property's hidden value



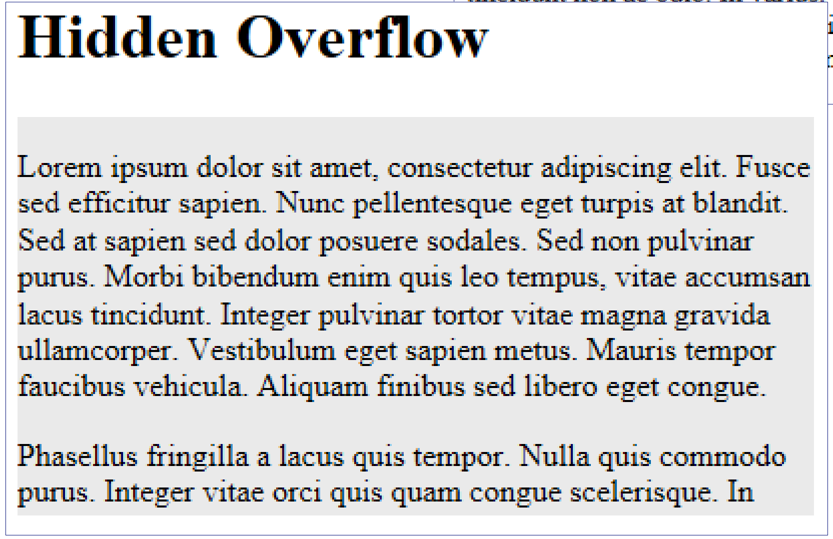
Visible Overflow

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce sed efficitur sapien. Nunc pellentesque eget turpis at blandit. Sed at sapien sed dolor posuere sodales. Sed non pulvinar purus. Morbi bibendum enim quis leo tempus, vitae accumsan lacus tincidunt. Integer pulvinar tortor vitae magna gravida ullamcorper. Vestibulum eget sapien metus. Mauris tempor faucibus vehicula. Aliquam finibus sed libero eget congue.

Phasellus fringilla a lacus quis tempor. Nulla quis commodo purus. Integer vitae orci quis quam congue scelerisque. In sodales augue tellus, id ullamcorper felis aliquet molestie. Fusce sodales semper augue id varius. Suspendisse lobortis cursus dolor eu tincidunt. Praesent et tortor a quam auctor tincidunt non ac odio. In varius, felis et molestie eleifend, ante

im ligula vel dui. Sed at

m cursus ex vel ullamcorper.



Hidden Overflow

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce sed efficitur sapien. Nunc pellentesque eget turpis at blandit. Sed at sapien sed dolor posuere sodales. Sed non pulvinar purus. Morbi bibendum enim quis leo tempus, vitae accumsan lacus tincidunt. Integer pulvinar tortor vitae magna gravida ullamcorper. Vestibulum eget sapien metus. Mauris tempor faucibus vehicula. Aliquam finibus sed libero eget congue.

Phasellus fringilla a lacus quis tempor. Nulla quis commodo purus. Integer vitae orci quis quam congue scelerisque. In

Overflow Demo

```
<body>
   <h1>Overflow Demo</h1>
  <p>Lorem ipsum dolor sit
  amet, consectetur
  adipiscing elit. Fusce sed
  efficitur sapien. Nunc
  pellentesque eget turpis at
  blandit. Sed at sapien sed
  dolor posuere sodales. Sed
  non pulvinar purus. Morbi
  bibendum enim quis leo
  tempus, vitae accumsan
  lacus tincidunt.</p>
</body>
```

```
<style>
    p {
       height: 200px;
       width: 200px;
       background-color: #e1e1e1;
       overflow: scroll | visible |
hidden;
    }
</style>
```

Summary

1	Cascading Style Sheets
2	Selectors and Declarations
3	Fonts and Font Families
4	Content Flow
5	Positioning
6	Content Overflow

Microsoft